B. TECH PROJECT

(COE – 415)

<u>Submitted By :</u>

KSHITIJ RAJPUT	276/CO/15
RAGHAV KAPOOR	322/CO/15

<u>Guided by :</u>

Mrs. PREETI KAUR
(Professor, NSUT)

DVESH PRAHARI :

ATTACKERS IN DISGUISE

Detecting Vindictive Behavior from Social Media during elections in Code-Switched Languages Using NLP and Vision Techniques

AIM

We aim to undertake a total of four individual tasks in order to achieve the final goal of labelling the pages and users on facebook and twitter which spring up during the election times.

Through the help of the proposed system, we would now be able to figure out the pages and users that misleadingly take part in the campaigns and create a fraudulent environment and misuse the freedom of speech.

These comprise of :-

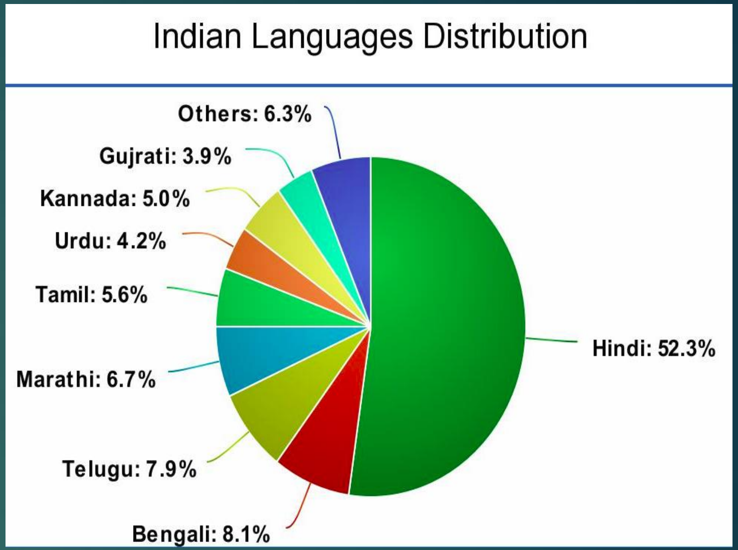
Creation of a suitable model and embeddings → Enhancing NLP Word2vec model → Deducing hate speech for images → Applying the work to various social media platforms

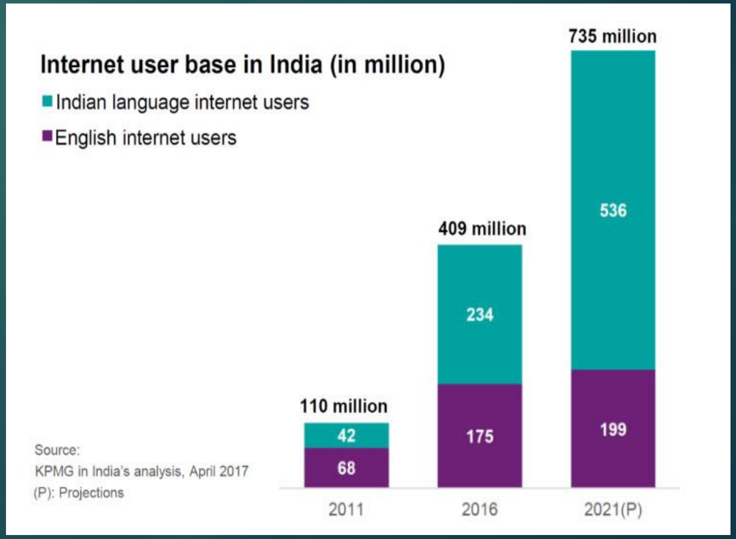
CURRENT SCENARIO

In multilingual societies (Eg: India), code-switched language(Hinglish) is most popular

Indian Internet Users crossed 500 million in 2018.

Tackles the difficulty of non-fixed grammar, vocabulary and semantics of the language pair.

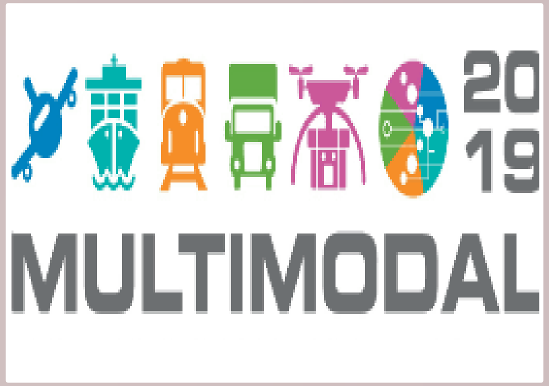




SCOPE OF WORK

- **Detect False Propaganda** by Political Groups in **Elections**
- **Youtube/ Netflix Subtitles –** "Auto-beep" offensive language
- **Online Social Media -** Report Defamatory Pages and comments
- **Feedback analytics** for better user experience.
- **Real time "clean-chat" facility.**
- **Censor board** – Auto-eliminate abusive content.

INTRODUCTION



Natural Language processing (NLP) is the analysis that explores how computers

perceive and manipulate

language

Hate speech is a statement intended to demean and brutalize another or the use of derogatory language

Mixed used of more than 2 languages in a text is called as code switching

For eg : Hindi-English (Hinglish)

Multimodality describes communication practices in terms of the textual, aural, linguistic, spatial, and visual resources

RELATED WORK

Sentiment Analysis

1. The task of sentiment analysis on code mixed Hi-En (Hinglish) social media content was first performed by Joshi et al. (2016) who used sub-word level representations in LSTM architecture to categorize tweets in positive, negative or neutral category.

2. Jhanwar et al. (2018) proposed an ensemble method for sentiment analysis on Hindi-English code switched dataset which outperformed the previous models of sentiment analysis.

3. Another method put forward by Gupta et al. (2018) performed the task of sentiment analysis on code switched Hinglish tweets dataset using a CNN based model consisting of a sentence representation matrix, convolution layer, pooling layer and fully connected layer

RELATED WORK

Hate Speech Detection

1. The initial task for hate speech detection was performed by Spertus et al. (1997), who developed a prototype system *Smokey* for detecting email flames (angry or offensive emails) using a 47 elements feature set which captured the syntax and semantics of the sentences present in the dataset.

2. The task of hate speech detection on Italian language using the following features: (i) morpho-syntactical features, (ii) sentiment polarity and (iii) word embedding lexicons was shown by Del Vigna (2017).

3. The task of hate speech detection on Hindi-English code switched data using a Random Forest (RF) classifier and a Support Vector Machine (SVM) classifier was performed by Bohra et al. (2018) using a 4 element feature set extracted from the tweets.

4. A ternary trans CNN model using transfer learning for hate speech detection on Hindi-English code switched dataset was proposed by Mathur et al. (2018).

RELATED WORK

Image Analysis

1. Research by Wang et al. (2006) focused on analyzing sentiment out of images by proposing a mechanism for finding the emotion out of an image by finding a orthogonal three dimensional factor space of a image and then passing it through a SVM classifier.

2. Siersdorfer et al. (2010) analyzed the relation between sentiment of images expressed in metadata and their visual content in the social photo sharing environment Flickr.

3. A progressive CNN model for visual sentiment analysis with transfer learning to learn the features on a twitter image dataset was also proposed by You et al. (2015).

4. Cai et al. (2017) performed sentiment analysis on the combination of text and images instead of considering them separately using two individual CNN architectures and them combining the results from both the architectures to calculate the final sentiment of the image.

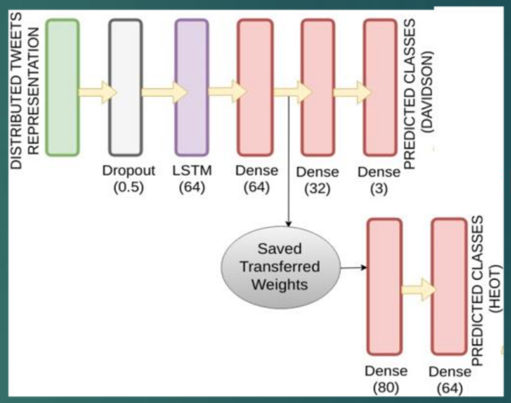
Classification of Tweets

WORKFLOW FOR TWEETS/ COMMENTS

हम उन्हें बर्बाद कर देंगे

ham unhe barbaad kar denge

we will destroy them

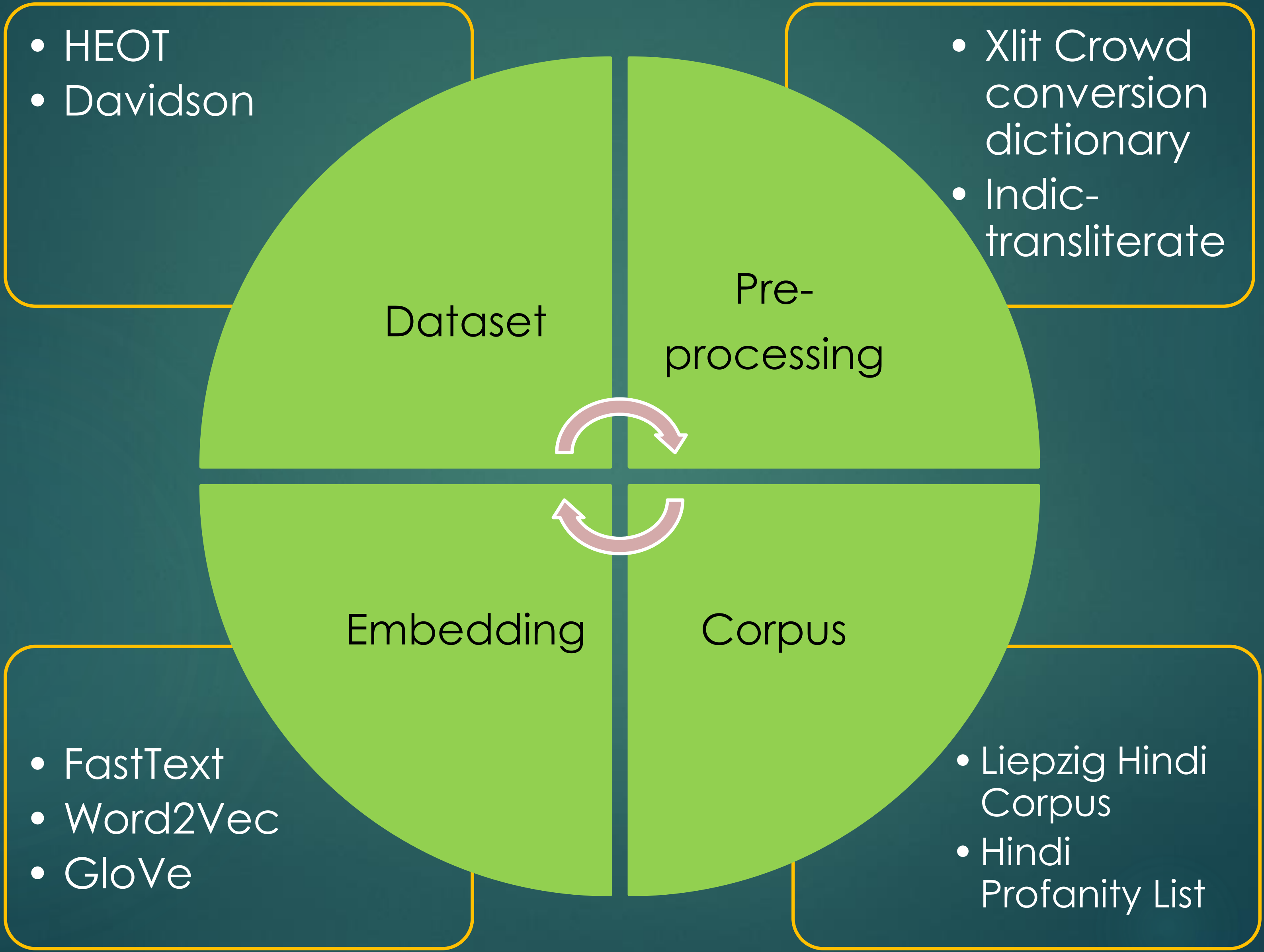


(Indic-transliteration python library)

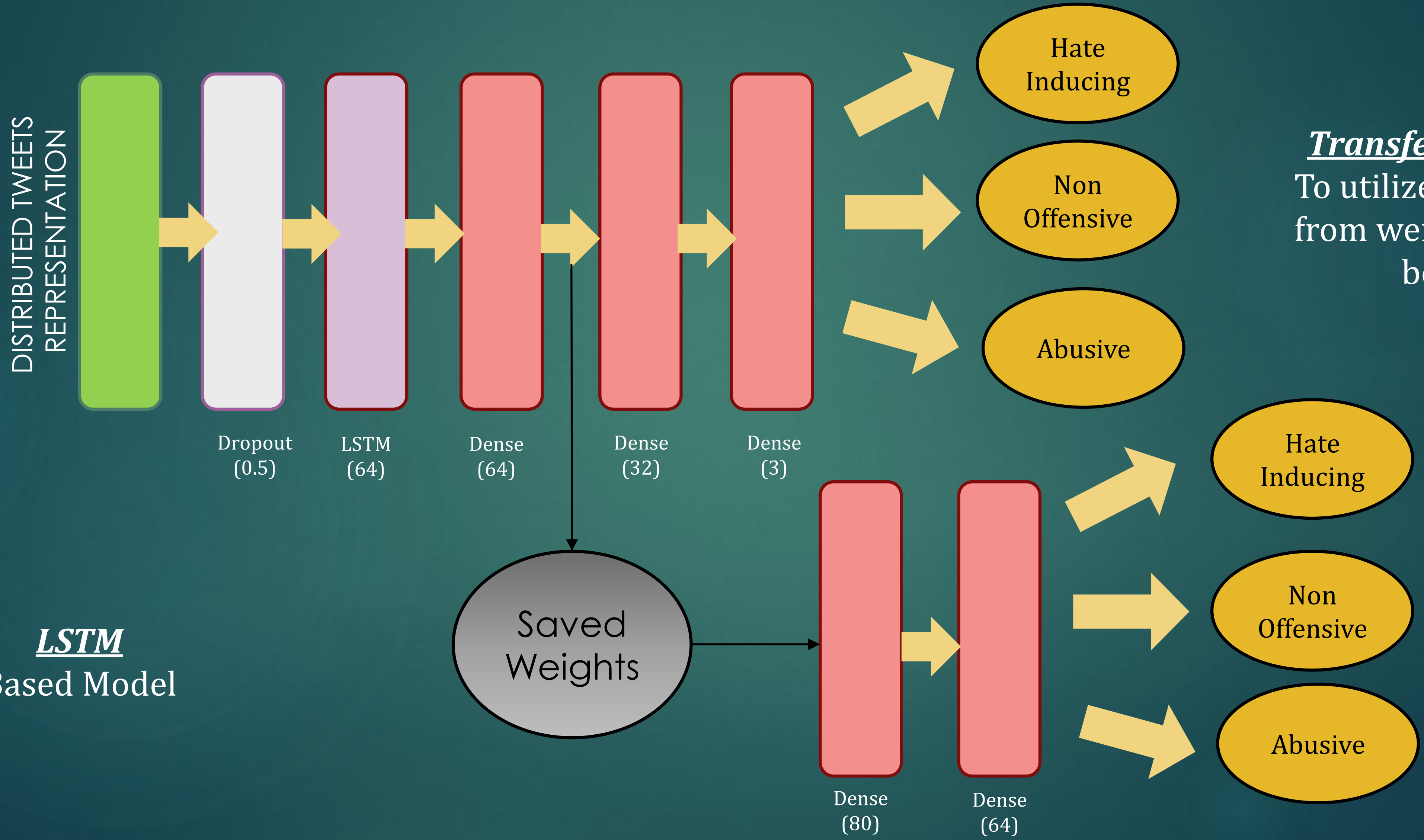
(Xlit Crowd Conversion Dictionary)

Comments from Social Media → Comments in Hindi → Trans-literation to English → Translation to English → Passes through our system → Highly Accurate Results

PREPROCESSING AND DATASET



MODEL - LSTM



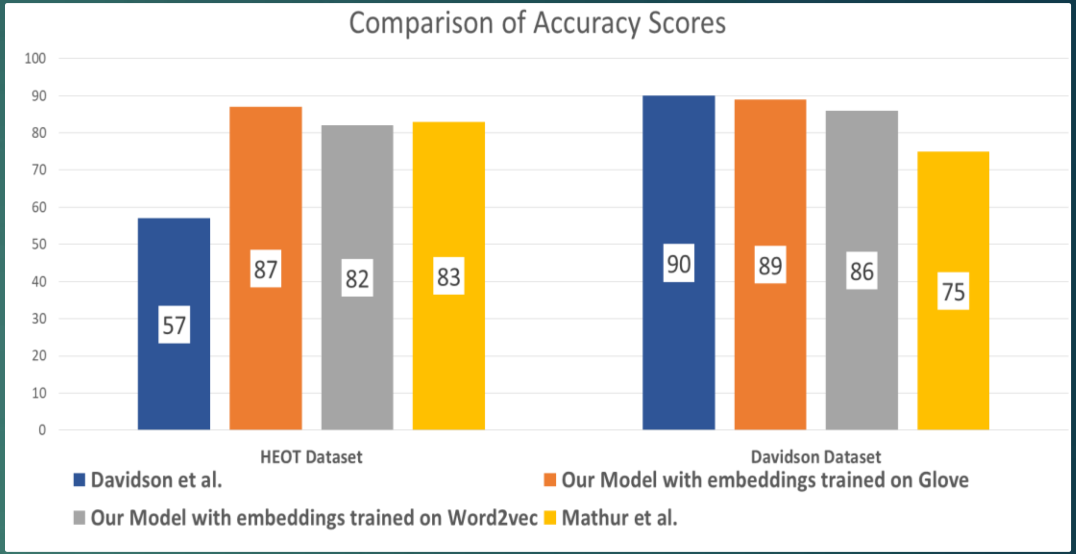
Transfer Learning
To utilize the benefits from weights trained before.

Ternary
classification model

LSTM
Based Model

TWEETS CLASSIFICATION RESULTS

State of the art results for Hinglish language.



- Accuracy, F1, Precision, Recall are used as metrics to evaluate the credibility of results.
- Best results for our model trained on Glove embeddings on HEOT dataset.
- Comparable results on Davidson English tweets dataset.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

MILESTONES ACHIEVED….

****** PUBLISHED IN PROCEEDINGS OF AAAI – 19, HONOLULU, HAWAII, U.S.A. *******
(Artificial Intelligence, Association for the Advancement of Artificial Intelligence.)
(H-INDEX – 69)
(A* Conference)

******* The Best Poster Award At AAAI-19 *******

****** SCHOLARSHIP FROM AAAI - 19 *******

DEPENDENCY BASED EMBEDDINGS

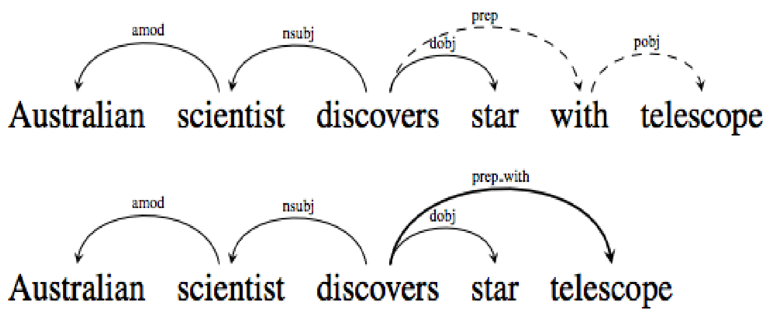
DEPENDENCY BASED EMBEDDINGS

Normal word2vec model considers adjacent words as a part of the context.

Parse Tree generated through dependency parser on ~10GB Hindi Corpus

Parse Tree passed to word2vecf to create unique set of embeddings called "*Context Based Embeddings*" which no more consider adjacent words

Smart Embeddings produce better results than Vanilla Embeddings



WORD	CONTEXTS
australian	scientist/amod^{-1}
scientist	australian/amod, discovers/nsubj^{-1}
discovers	scientist/nsubj, star/dobj, telescope/prep_with
star	discovers/dobj^{-1}
telescope	discovers/prep_with^{-1}

Embedding dimension used was 50, 100, 300, 500

Context window size used was 2,3,5

WORD EMBEDDINGS

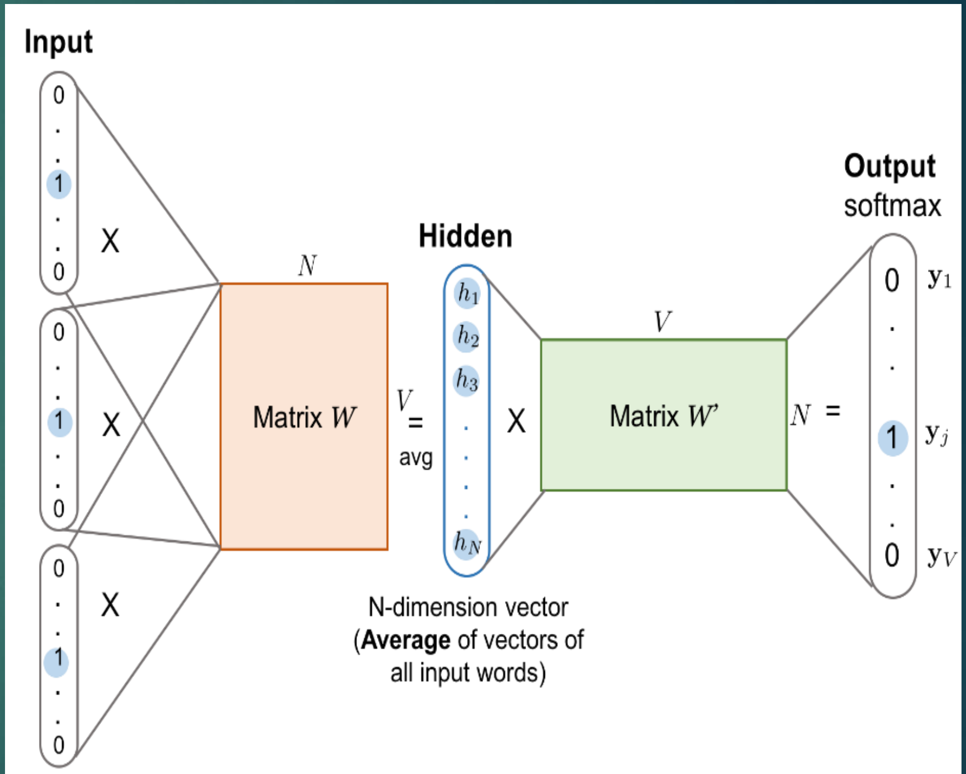
Word embedding is the name for a set in NLP where words from the vocabulary are mapped to vectors.

These are used to form the first layer of our model.

These embeddings help to learn the distributed representations of tweets by creating word vectors

We have used 3 embeddings for this task :

- CBOW Word2Vec
- Skip Gram Word2Vec
- Dependency based embeddings

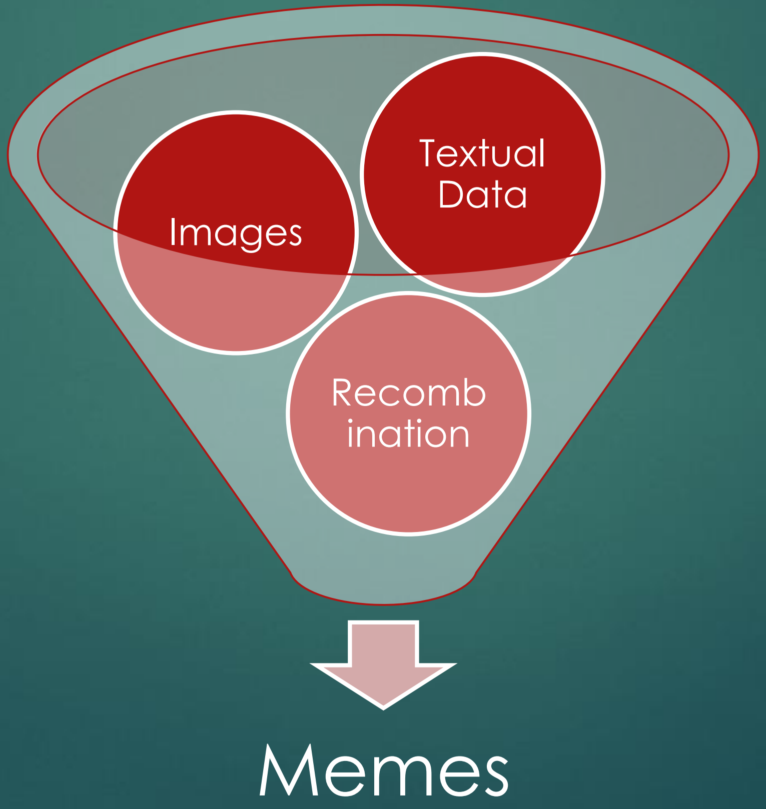


DEPENDENCY BASED RESULTS

- Dependency based embeddings perform at par than vanilla embeddings on LSTM model on HEOT dataset
- Huge rise of 6% in Skip Gram based word2Vec embeddings
- Massive shoot up of 7% in CBOW word2Vec embeddings

Embeddings				Accuracy	Precision
Training Algorithm	Window Size	Vector Size	Type		
SG	2	100	Vanilla	82.56 %	0.81
SG	**2**	**100**	**Dependency**	**88.14 %**	**0.89**
CBOW	2	100	Vanilla	79.68 %	0.80
CBOW	**2**	**100**	**Dependency**	**86.01 %**	**0.88**

CLASSIFICATION OF IMAGES



INDIAN POLITICAL MEMES (IPM) DATASET

- A dataset of 1218 images which are shared by politically inclined users on social media will be scraped using *Google_image_download*.
- Triply annotated dataset in 3 categories - hate-inducing, benign and satirical content.
- This dataset is used for multimodal hate speech classification, i.e., for images / Memes analysis.

Hate-Inducing

Benign

Satirical

INDIAN POLITICAL MEMES (IPM) DATASET

Figure	Hinglish Text Extracted	English Translation	Label
Figure 1	Udi baba, Mark, homse kab milega ?	Hey Mark, when will you meet us ?	Non Offensive
Figure 2	Kisne kaha ki main pogo dekhta hun. Mummy se shikayat karunga	Who said that I watch pogo. I will complain to mother	Satirical
Figure 3	Ye to acha hai India mein beauty contest mein reservation nhi hoti.	It is good that there in no reservation in beauty contests in India (Derogatory remark on personal appearance)	Hate Inducing

The validation of dataset was done on 3 parameters :

- **Cohen's Kappa** – Denotes the inter annotator agreement
  Highest Cohen kappa was recorded 0.87, which signifies high agreement

$$k \equiv \frac{p_0 - p_e}{1 - p_e} \equiv 1 - \frac{1 - p_o}{1 - p_e}$$

INDIAN POLITICAL MEMES (IPM) DATASET

- **Fleiss's Kappa** – Denotes the inter annotator agreement of annotators greater than 2.
  This was calculated as 0.782. The dataset is annotated with great agreement factor

$$k = \frac{p_a - p_e}{1 - p_e}$$

- **Mulitlingual Index (MI)** : Denoted the degree of mixture of languages present in the dataset.
  - The value for IPM dataset was 0.684 indicating a balanced dataset and equal proportions of English and hinglish.

$$MI = \frac{1 - \sum_{j=1}^{k} p_j}{(k-1)\sum_{j=1}^{k} p_j^2}$$

WORKFLOW FOR IMAGES / MEMES

MULTIMODAL HATE SPEECH IDENTIFICATION – Analysis of Text and Images in parallel using binary channel CNN-LSTM based model

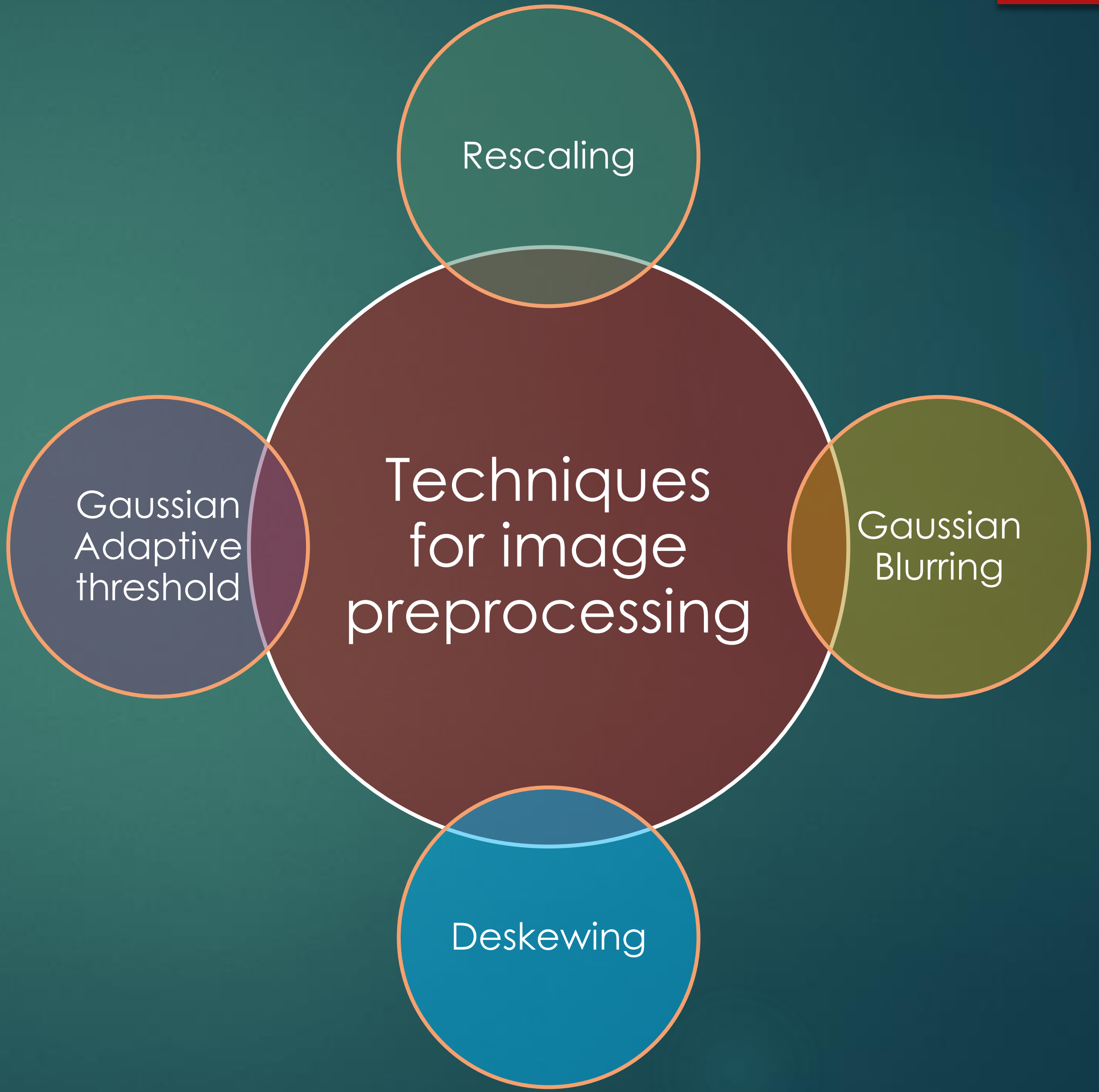
- OCR Extraction of text from Images
- Use Text from Image captions
- Use CNN-LSTM model for image classification. LSTM for text and CNN for images.
- Recombination of the two channels
- Final classification of Memes on IPM dataset

PREPROCESSING IMAGES

- The image needs to be pre-processed in order to extract text out of the meme efficiently.

- The OCR reader performs some of the image preprocessing

- We did some manual preprocessing which includes the following techniques



DATA AUGMENTATION

- Data augmentation refers to methods for constructing iterative optimization or sampling algorithms via the introduction of unobserved data or latent variables

- The technique of data augmentation is used to increase the size of the dataset to train the classifier model

- We have used mainly 5 different techniques of data augmentation :

Scaling

Translation

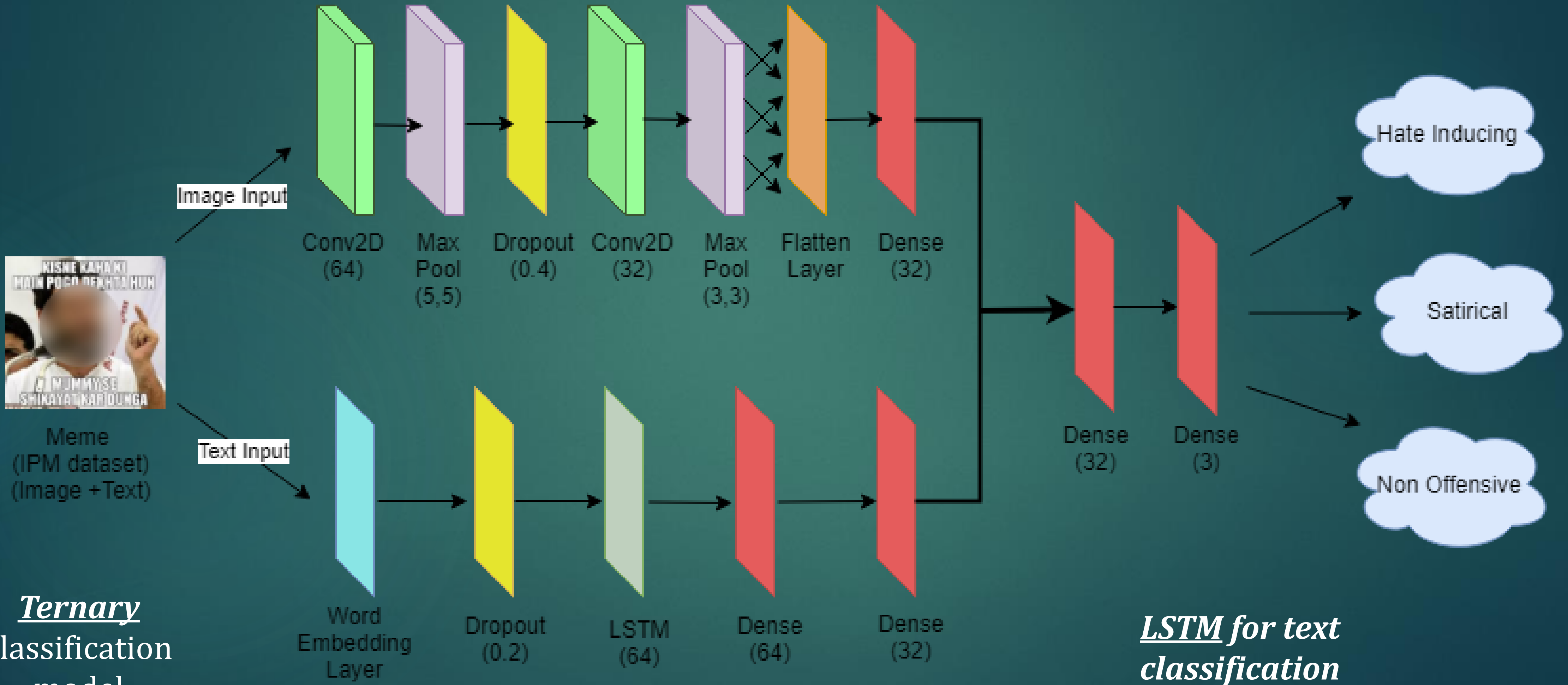
Rotation

Adding Noise

Flipping

MODEL

CNN ***for image classification***



Ternary classification model

LSTM ***for text classification***

EXPERIMENTATION

Baseline Models

Support Vector Machine (SVM)

- Kernel = ‘poly’
- Degree = 3
- All other hyper parameters are set to default

Random Forest (RF)

- N_estimators = 600
- Max_depth = 12
- Max_features = *log2*

CNN model

- Convolution2D layer of filter size = 64 & kernel size = (5, 5).
- Max_pooling layer of pool size = (5, 5).
- Dense layer size = 64 and a Dropout layer= 0.4

LSTM model

- LSTM layer of size = 64
- Adam optimizer and L2 regularization.
- Dropout layer = 0.4

EXPERIMENTATION

Features Extracted for input to SVM and RF models

GLCM

- Gray-Level Co-occurrence Matrix (GLCM) gives the texture of the image which is useful for determining the emotional expression in an image

Color

- Calculated using Earth Mover's Distance (EMD) between the histogram of an image and the histogram having a uniform color distribution
- Color can be the means of spreading religious hatred through a meme.

Tamura features

- Tamura features capture the texture of an image more effectively.

Human Face

- Human face feature is used because two similar images can have different impact on the user depending on the fact that it contains a face or not

MEMES CLASSIFICATION RESULTS

Features	Precision	Recall	F1-Score
Glove(Gl)	**0.812**	**0.816**	**0.822**
Twitter Word2Vec(Tw)	0.781	0.786	0.781
FastText(Ft)	0.791	0.784	0.793
Bert(Bt)	0.758	0.804	0.784
(Gl) + (Tw)	0.727	0.751	0.722
(Gl) + (Ft)	**0.803**	**0.811**	**0.810**
(Bt) + (Tw)	0.788	0.780	0.772
(Gl) + (Ft)	0.779	0.790	0.765
(Bt) + (Ft)	0.780	0.783	0.796

State of the art results for Image Classification on IPM dataset

We use CNN – LSTM based model with different flavours of embeddings : Word2Vec, Glove, Fastext, Bert

Comparison with other models :

- SVM – 19%
- RF – 23%
- CNN – 17%
- LSTM – 22%

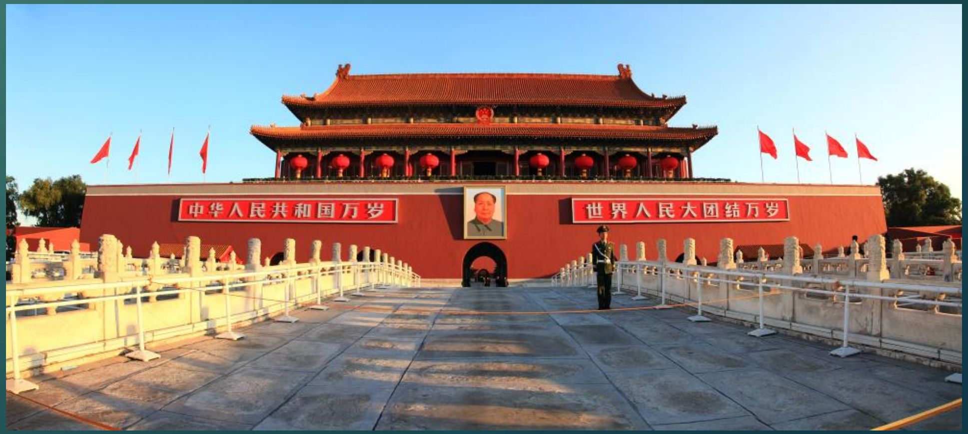
MILESTONES ACHIEVED….

****** ANOTHER PAPER SUBMITTED IN
28TH ACM CONFERENCE ON INFORMATION RETRIEVAL AND KNOWLEDGE MANAGEMENT
BEIJIING, CHINA ******

***** PAPER TITLE - HATE ME NOT : DETECTING HATE INDUCING MEMES IN CODE SWITCHED LANGUAGES *****

****** A+ CONFERENCE WITH H-INDEX – 42 *******

CIKM2019 | BEIJING CHINA
November 03-07. 2019



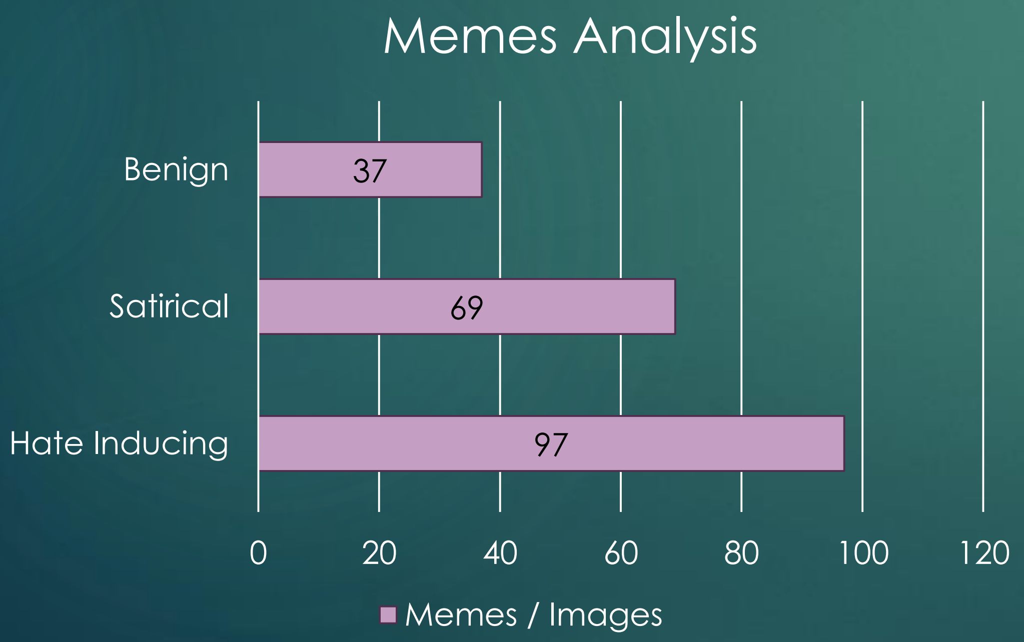
APPLICATION ON SOCIAL MEDIA

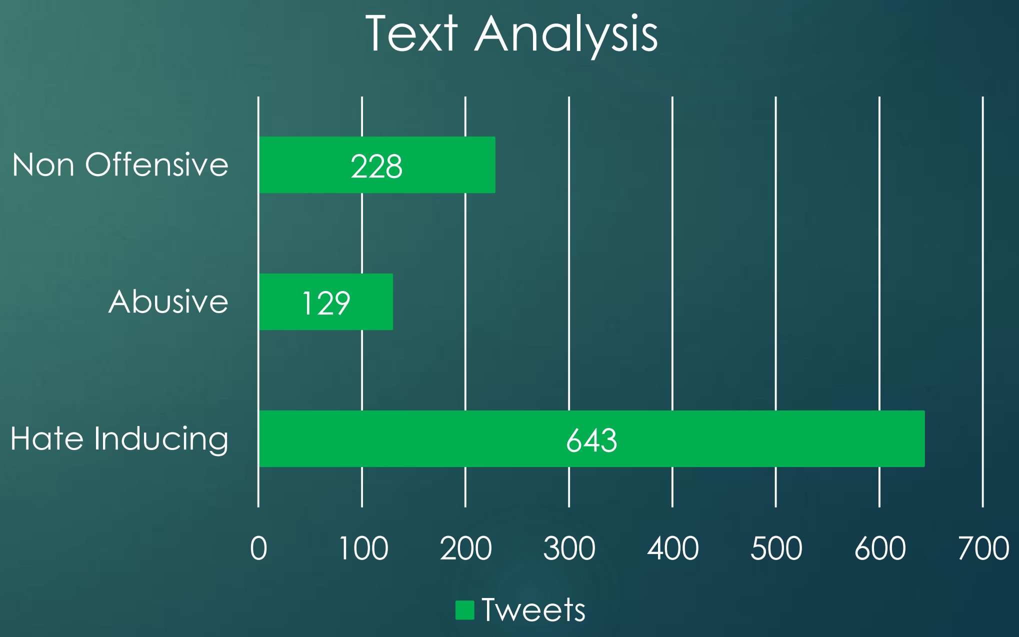
WORKFLOW MODEL

1. Scrape Tweets from twitter handle using Twitter API
2. Scrape Images from Twitter handle by using twitter-photos python package
3. Preprocess and classify tweets using the LSTM model
4. Process and Classify images using CNN-LSTM memes model
5. Plotted Graphs using matplotlib to analyse the pages and twitter handles

RESULTS - @theskindoctor13

- ✓ We extracted 1000 tweets and 200 images from twitter handle - **@theskindoctor13**
- ✓ 48.5 % Memes are Hate inducing and 64.3 % Tweets are Hate Inducing
- ✓ Suggesting the page is hate inducing and must be removed fro Twitter





Conclusion

- Prevent vindictive behaviour on social media
- Curb misuse of freedom of speech online
- Proposed a LSTM based model for textual data.
- Used dependency based embeddings to improve results
- Created a Indian Political Memes Dataset(IPMD)
- Multimodal hate and abusive speech detection
- Identify pages and users that spread false propaganda online.

APPLICATIONS

Use by whatsapp, messenger to remove bad words

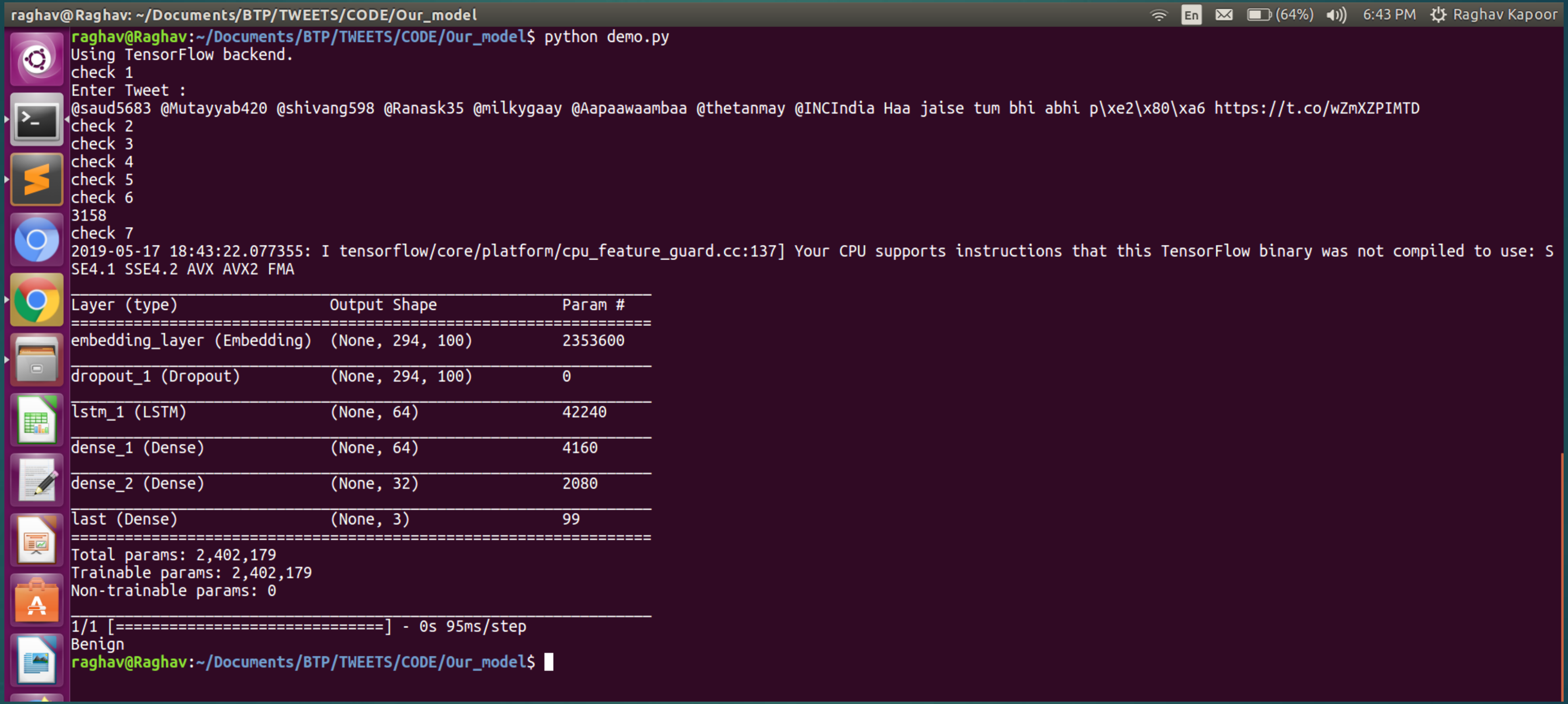
Online Social Media - Report Defamatory Pages and comments

Video streaming applications use for abuse free subtitles

User Feedback on various platforms to analyse sentiments of users

Election Commission use for detecting false propaganda

Demo Screenshots

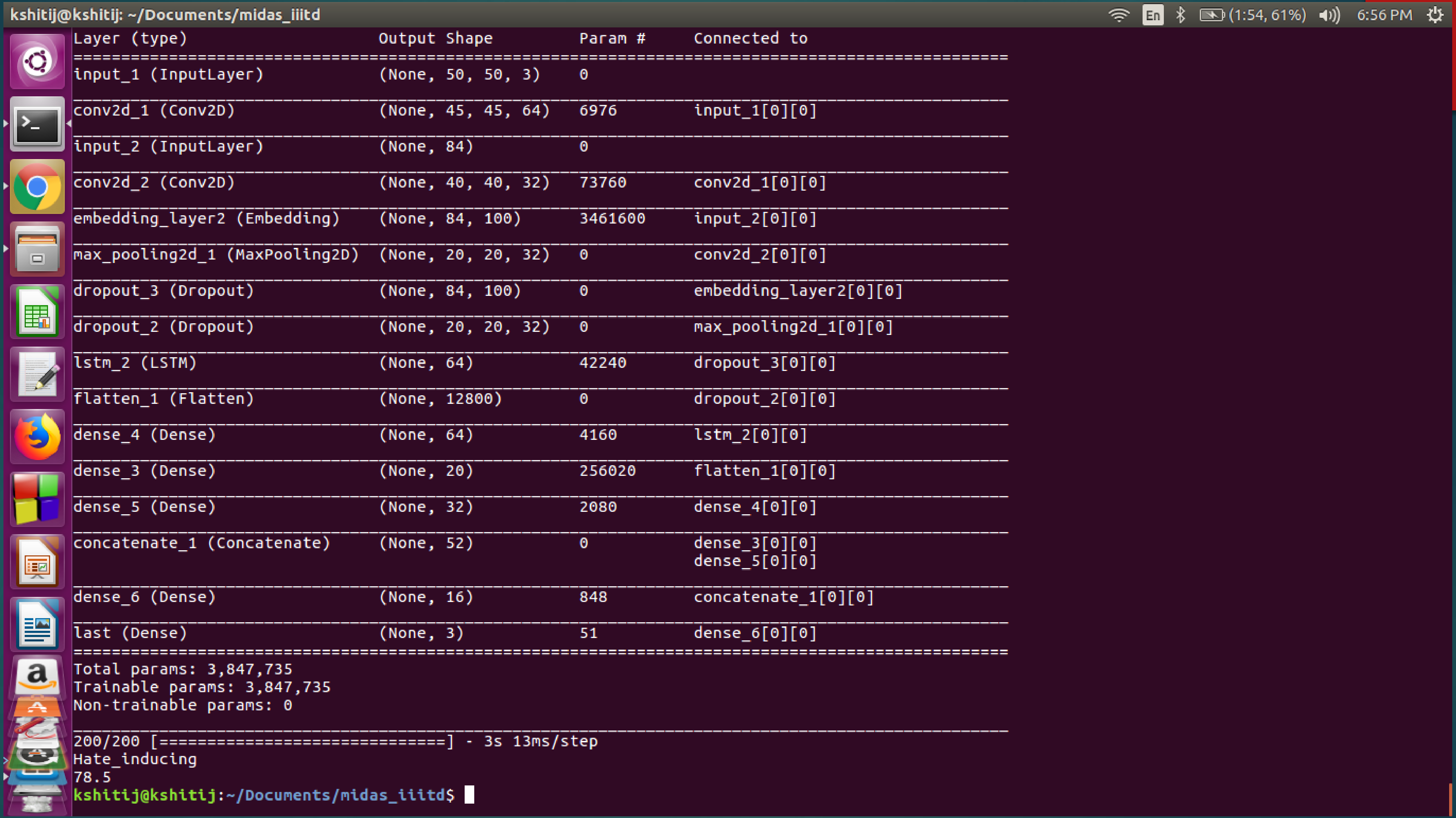


Tweets classification Demo

```
TeamViewer 14
2502/2502 [==============================] - 23s 9ms/step - loss: 0.0621 - acc: 0.9824
Epoch 23/25
2502/2502 [==============================] - 27s 11ms/step - loss: 0.0574 - acc: 0.9776
Epoch 24/25
2502/2502 [==============================] - 24s 10ms/step - loss: 0.0556 - acc: 0.9800
Epoch 25/25
2502/2502 [==============================] - 22s 9ms/step - loss: 0.0479 - acc: 0.9832
656/656 [==============================] - 2s 3ms/step
656/656 [==============================] - 2s 3ms/step
hello
84.9085365854
84.9085365854
0.84875991588
0.850222137386
0.849085365854
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_layer (Embedding)  (None, 294, 100)          2353600
_________________________________________________________________
dropout_1 (Dropout)          (None, 294, 100)          0
_________________________________________________________________
lstm_1 (LSTM)                (None, 64)                42240
_________________________________________________________________
dense_1 (Dense)              (None, 64)                4160
=================================================================
Total params: 2,400,000
Trainable params: 0
Non-trainable params: 2,400,000
_________________________________________________________________
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_layer (Embedding)  (None, 294, 100)          2353600
_________________________________________________________________
dropout_1 (Dropout)          (None, 294, 100)          0
_________________________________________________________________
lstm_1 (LSTM)                (None, 64)                42240
_________________________________________________________________
dense_1 (Dense)              (None, 64)                4160
_________________________________________________________________
dense_3 (Dense)              (None, 80)                320
_________________________________________________________________
last_again (Dense)           (None, 3)                 243
=================================================================
Total params: 2,400,563
Trainable params: 563
```

```
TeamViewer 14
Epoch 1/20
2502/2502 [==============================] - 6s 3ms/step - loss: 0.9674 - acc: 0.6998
Epoch 2/20
2502/2502 [==============================] - 6s 2ms/step - loss: 0.6166 - acc: 0.9892
Epoch 3/20
2502/2502 [==============================] - 9s 4ms/step - loss: 0.3648 - acc: 0.9876
Epoch 4/20
2502/2502 [==============================] - 6s 2ms/step - loss: 0.2157 - acc: 0.9872
Epoch 5/20
2502/2502 [==============================] - 8s 3ms/step - loss: 0.1372 - acc: 0.9868
Epoch 6/20
2502/2502 [==============================] - 6s 3ms/step - loss: 0.1080 - acc: 0.9848
Epoch 7/20
2502/2502 [==============================] - 7s 3ms/step - loss: 0.0813 - acc: 0.9864
Epoch 8/20
2502/2502 [==============================] - 6s 2ms/step - loss: 0.0637 - acc: 0.9888
Epoch 9/20
2502/2502 [==============================] - 6s 2ms/step - loss: 0.0579 - acc: 0.9900
Epoch 10/20
2502/2502 [==============================] - 6s 2ms/step - loss: 0.0553 - acc: 0.9884
Epoch 11/20
2502/2502 [==============================] - 6s 2ms/step - loss: 0.0566 - acc: 0.9868
Epoch 12/20
2502/2502 [==============================] - 6s 3ms/step - loss: 0.0651 - acc: 0.9840
Epoch 13/20
2502/2502 [==============================] - 8s 3ms/step - loss: 0.0491 - acc: 0.9868
Epoch 14/20
2502/2502 [==============================] - 6s 2ms/step - loss: 0.0506 - acc: 0.9868
Epoch 15/20
2502/2502 [==============================] - 6s 2ms/step - loss: 0.0503 - acc: 0.9872
Epoch 16/20
2502/2502 [==============================] - 6s 2ms/step - loss: 0.0497 - acc: 0.9884
Epoch 17/20
2502/2502 [==============================] - 6s 2ms/step - loss: 0.0496 - acc: 0.9876
Epoch 18/20
2502/2502 [==============================] - 6s 2ms/step - loss: 0.0545 - acc: 0.9868
Epoch 19/20
2502/2502 [==============================] - 8s 3ms/step - loss: 0.0440 - acc: 0.9884
Epoch 20/20
2502/2502 [==============================] - 7s 3ms/step - loss: 0.0548 - acc: 0.9852
656/656 [==============================] - 2s 3ms/step
hello again
85.0609756098
0.85016941768
0.851325094574
0.850609756098
raghav@Raghav:~/Documents/BTP/TWEETS/CODE/Our_model$
```

Text Classification Model



```
kshitij@kshitij: ~/Documents/midas_iiitd
Layer (type)                     Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)             (None, 50, 50, 3)    0
__________________________________________________________________________________________________
conv2d_1 (Conv2D)                (None, 45, 45, 64)   6976        input_1[0][0]
__________________________________________________________________________________________________
input_2 (InputLayer)             (None, 84)           0
__________________________________________________________________________________________________
conv2d_2 (Conv2D)                (None, 40, 40, 32)   73760       conv2d_1[0][0]
__________________________________________________________________________________________________
embedding_layer2 (Embedding)     (None, 84, 100)      3461600     input_2[0][0]
__________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)   (None, 20, 20, 32)   0           conv2d_2[0][0]
__________________________________________________________________________________________________
dropout_3 (Dropout)              (None, 84, 100)      0           embedding_layer2[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)              (None, 20, 20, 32)   0           max_pooling2d_1[0][0]
__________________________________________________________________________________________________
lstm_2 (LSTM)                    (None, 64)           42240       dropout_3[0][0]
__________________________________________________________________________________________________
flatten_1 (Flatten)              (None, 12800)        0           dropout_2[0][0]
__________________________________________________________________________________________________
dense_4 (Dense)                  (None, 64)           4160        lstm_2[0][0]
__________________________________________________________________________________________________
dense_3 (Dense)                  (None, 20)           256020      flatten_1[0][0]
__________________________________________________________________________________________________
dense_5 (Dense)                  (None, 32)           2080        dense_4[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)      (None, 52)           0           dense_3[0][0]
                                                                  dense_5[0][0]
__________________________________________________________________________________________________
dense_6 (Dense)                  (None, 16)           848         concatenate_1[0][0]
__________________________________________________________________________________________________
last (Dense)                     (None, 3)            51          dense_6[0][0]
==================================================================================================
Total params: 3,847,735
Trainable params: 3,847,735
Non-trainable params: 0
__________________________________________________________________________________________________
200/200 [==============================] - 3s 13ms/step
Hate_inducing
78.5
kshitij@kshitij:~/Documents/midas_iiitd$
```

Memes Classification Model and Demo

MORE ABOUT OUR WORK

Link to the Paper:

https://www.aaai.org/Papers/AAAI/2019/SA-KapoorR.211.pdf

Link to Project Video :

https://drive.google.com/file/d/1z_FCuaBQiYRvD6LrV8g4_xr2x5Y57huQ/view?usp=sharing

Link to Award Winning Poster :

https://drive.google.com/file/d/1oGTknE0_sGJVMS9FUk52Nf9ZIzgJQ4cm/view?usp=sharing

REFERENCES

[1] K. Bali, J. Sharma, M. Choudhury, and Y. Vyas, "' I am borrowing ya mixing?' An Analysis of English-Hindi Code Mixing in Facebook," in *Proceedings of the First Workshop on Computational Approaches to Code Switching*, 2014, pp. 116–126.

[2] A. Das and B. Gambäck, "Identifying languages at the word level in code-mixed indian social media text," in *Proceedings of the 11th International Conference on Natural Language Processing*, 2014, pp. 378–387.

[3] D. Mave, S. Maharjan, and T. Solorio, "Language Identification and Analysis of Code-Switched Social Media Text," in *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, 2018, pp. 51–61.

[4] K. Singh, I. Sen, and P. Kumaraguru, "Language identification and named entity recognition in hinglish code mixed tweets," in *Proceedings of ACL 2018, Student Research Workshop*, 2018, pp. 52–58.

[5] Y. Vyas, S. Gella, J. Sharma, K. Bali, and M. Choudhury, "Pos tagging of english-hindi code-mixed social media content," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 974–979.

[6] D. Gupta, S. Tripathi, A. Ekbal, and P. Bhattacharyya, "SMPOST: parts of speech tagger for code-mixed Indic social media text," *arXiv preprint arXiv:1702. 00167*, 2017.

[7] P. R. K. Rao and S. L. Devi, "CMEE-IL: Code Mix Entity Extraction in Indian Languages from Social Media Text@ FIRE 2016-An Overview," in *FIRE (Working Notes)*, 2016, pp. 289–295.

[8] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

REFERENCES

[9] A. Prabhu, A. Joshi, M. Shrivastava, and V. Varma, "Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text," *arXiv preprint arXiv:1611. 00472*, 2016.

[10] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2*, 2012, pp. 90–94.

[11] B. Pang, L. Lee, and Others, "Opinion mining and sentiment analysis," *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.

[12] S. Sharma, P. Srinivas, and R. C. Balabantaray, "Text normalization of code mix and sentiment analysis," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2015, pp. 1468–1473.

[13] M. G. Jhanwar and A. Das, "An Ensemble Model for Sentiment Analysis of Hindi-English Code-Mixed Data," *arXiv preprint arXiv:1806.04450*, 2018.

[14] D. Gupta, A. Lamba, A. Ekbal, and P. Bhattacharyya, "Opinion Mining in a Code-Mixed Environment: A Case Study with Government Portals," in *Proceedings of the 13th International Conference on Natural Language Processing*, 2016, pp. 249–258.

[15] E. Spertus, "Smokey: Automatic recognition of hostile messages," in *Aaai/iaai*, 1997, pp. 1058–1065.

[16] D. Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis, and L. Edwards, "Detection of harassment on web 2.0," *Proceedings of the Content Analysis in the WEB*, vol. 2, pp. 1–7, 2009.

[17] S. O. Sood, E. F. Churchill, and J. Antin, "Automatic identification of personal insults on social news sites," *J. Am. Soc. Inf. Sci. Technol.*, vol. 63, no. 2, pp. 270–285, 2012.

REFERENCES

[18] G. Xiang, B. Fan, L. Wang, J. Hong, and C. Rose, “Detecting offensive tweets via topical feature discovery over a large scale twitter corpus,” in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 1980–1984.

[19] F. Del Vigna12, A. Cimino23, F. Dell’Orletta, M. Petrocchi, and M. Tesconi, “Hate me, hate me not: Hate speech detection on Facebook,” 2017.

[20] A. Bohra, D. Vijay, V. Singh, S. S. Akhtar, and M. Shrivastava, “A Dataset of Hindi-English Code-Mixed Social Media Text for Hate Speech Detection,” in *Proceedings of the Second Workshop on Computational Modeling of People’s Opinions, Personality, and Emotions in Social Media*, 2018, pp. 36–41.

[21] P. Mathur, R. Shah, R. Sawhney, and D. Mahata, “Detecting offensive tweets in hindi-english code-switched language,” in *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, 2018, pp. 18–26.

[22] P. Mathur, R. Sawhney, M. Ayyar, and R. Shah, “Did you offend me? classification of offensive tweets in hinglish language,” in *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, 2018, pp. 138–148.

[23] W. Wei-ning, Y. Ying-lin, and J. Sheng-ming, “Image retrieval by emotional semantics: A study of emotional space and feature extraction,” in *2006 IEEE International Conference on Systems, Man and Cybernetics*, 2006, vol. 4, pp. 3534–3539.

[24] S. Siersdorfer, E. Minack, F. Deng, and J. Hare, “Analyzing and predicting sentiment of images on the social web,” in *Proceedings of the 18th ACM international conference on Multimedia*, 2010, pp. 715–718.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

REFERENCES

[26] Q. You, J. Luo, H. Jin, and J. Yang, "Robust image sentiment analysis using progressively trained and domain transferred deep networks," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[27] G. Cai and B. Xia, "Convolutional neural networks for multimedia sentiment analysis," in *Natural Language Processing and Chinese Computing*, Springer, 2015, pp. 159–167.

[28] S. Poria, I. Chaturvedi, E. Cambria, and A. Hussain, "Convolutional MKL based multimodal emotion recognition and sentiment analysis," in *2016 IEEE 16th international conference on data mining (ICDM)*, 2016, pp. 439–448.

[29] L.-P. Morency, R. Mihalcea, and P. Doshi, "Towards multimodal sentiment analysis: Harvesting opinions from the web," in *Proceedings of the 13th international conference on multimodal interfaces*, 2011, pp. 169–176.

[30] R. Kapoor, Y. Kumar, K. Rajput, R. R. Shah, P. Kumaraguru, and R. Zimmermann, "Mind Your Language: Abuse and Offense Detection for Code-Switched Languages," *arXiv preprint arXiv:1809. 08652*, 2018.

[31] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[32] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, 1960.

[33] R. Barnett *et al.*, "The LIDES Coding Manual: A Document for Preparing and Analyzing Language Interaction Data Version 1.1--July 1999," *International Journal of Bilingualism*, vol. 4, no. 2, pp. 131–271, 2000.

[34] J. L. Fleiss, "Measuring nominal scale agreement among many raters," *Psychol. Bull.*, vol. 76, no. 5, p. 378, 1971.

REFERENCES

[35] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[36] F. Godin, B. Vandersmissen, W. De Neve, and R. Van de Walle, "Multimedia Lab @ ACL WNUT NER Shared Task: Named Entity Recognition for Twitter Microposts using Distributed Word Representations," in *Proceedings of the Workshop on Noisy User-generated Text*, 2015, pp. 146–153.

[37] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.," 2008.

[38] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, 1999, vol. 2, pp. 246–252.

[39] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[40] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810. 04805*, 2018.

[41] D. A. Van Dyk and X.-L. Meng, "The art of data augmentation," *J. Comput. Graph. Stat.*, vol. 10, no. 1, pp. 1–50, 2001.

[42] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412. 6980*, 2014.

THANK YOU